Intelligent NoC with Neuro-Fuzzy Bandwidth Regulation for a 51 IP Object Recognition Processor

Seungjin Lee, Jinwook Oh, Minsu Kim, Junyoung Park, Joonsoo Kwon, Joo-Young Kim, and Hoi-Jun Yoo
Department of Electrical Engineering
KAIST
Daejeon, Republic of Korea
seungjin@eeinfo.kaist.ac.kr

***Abstract*—Balancing the execution times of concurrent tasks in a multi-core processor is critical to achieving good performance scaling with increasing core count. However, this is difficult when the tasks' execution times are not known in advance. In this work, we propose an intelligent Network-on-Chip that performs bandwidth regulation using weighted round robin packet arbitration to balance the execution times of 4 Feature Extraction Clusters whose workloads vary depending on the input content. A neuro-fuzzy inference block, named the Intelligent Inference Engine, predicts the workload of each FEC, and assigns a priority weight to each FEC channel. As a result, 34% reduction in synchronization overhead due to unbalanced execution time was achieved, and the overall execution time was reduced by 11.5%.**

I. INTRODUCTION

Embedded applications such as automobile driver assistance and augmented reality require parallel multi-core object recognition processors to achieve real-time performance while consuming low power [1,2]. The power efficiency of an object recognition processor can be further improved by visual attention, which selects only relevant image regions for detailed processing. Object recognition processors containing dedicated accelerators for visual attention [3,4] have been shown to reduce power consumption for certain scenes. However, their limitation is that they employ only bottom-up saliency based attention [5], which only works when the target object is more salient than the background.

The Unified Visual Attention Model (UVAM) [6] combines saliency based attention with top-down familiarity attention to improve attention accuracy, and thus achieve average workload reduction of 76%, even for scenes with salient backgrounds. An important feature of the UVAM is the feedback path from the recognition result back to attention, which guides the subsequent selection of image regions towards target objects. However, this introduces dependencies between the otherwise parallel object recognition tasks, which may negatively impact the utilization of a multi-core processor.

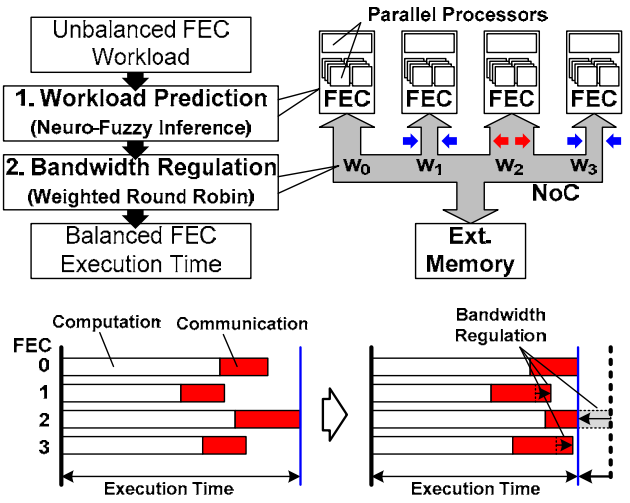


Figure 1. Execution time balancing by bandwidth regulation.

Dependencies between parallel tasks in a multi-core processor introduce synchronization issues. Most often, a synchronization barrier [7] is used to block execution of each task until all tasks arrive at a certain point in the algorithm. It is therefore crucial to balance the execution time of parallel tasks in order to minimize the idle time, and thus achieve high application performance. The two main factors that cause unbalanced execution time of parallel tasks are their dependency on the input data and/or algorithm and their dependency on external resources, such as shared memory bandwidth.

In this work, we present an intelligent Network-on-Chip (NoC) to balance the execution time of concurrent tasks in an heterogeneous multi-core object recognition processor [8] through dynamic workload prediction and bandwidth regulation. The workload prediction is performed efficiently by the Intelligent Inference Engine (IIE), a mixed-mode neuro-fuzzy logic block. Based on this prediction, the uneven execution times of the tasks can be balanced out using bandwidth regulation implemented by weighted round robin arbitration[9].

978-1-4244-5760-1/10/$26.00 ©2010 IEEE

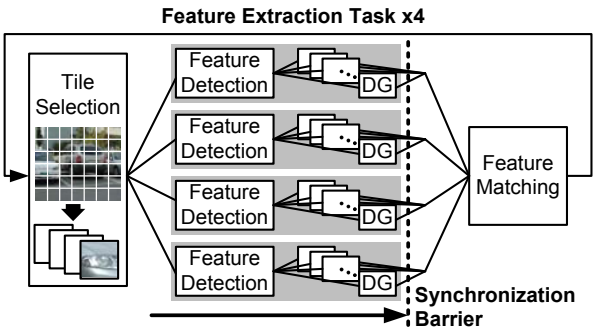


Figure 2. Task diagram of UVAM based object recognition.

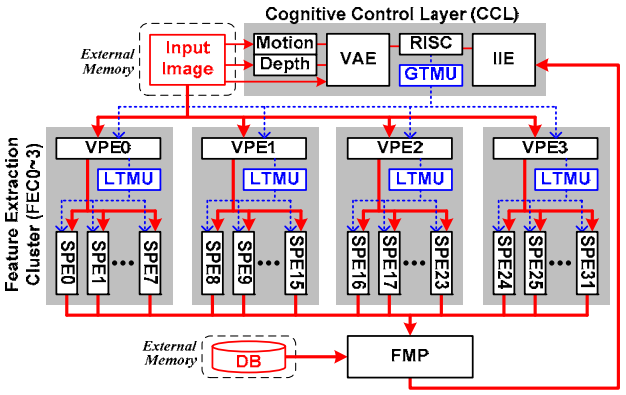


Figure 3. UVAM hardware mapping.

II. Target Application

A. UVAM based Object Recognition

The Unified Visual Attention Model (UVAM) [6], combines bottom-up saliency attention and top-down familiarity attention to attain highly accurate visual attention selection. The key advantage of the UVAM is that the result of feature extraction and subsequent feature matching on previously selected tiles is used to select the next set of tiles, as depicted by the feedback loop in fig. 2. As a result, top-down knowledge of the target objects is incorporated into the tile selection process, leading to more accurate tile selection and lower overall workload.

The most computationally complex portion of the UVAM is the feature extraction (FE) task, composed of feature detection (FD) and descriptor generation (DG) tasks. Each FE task takes 1 32x32 pixel image tile as its input, and outputs 1 128 dimensional descriptor vector for each feature detected in that tile. Despite its high complexity, the FE task can be accelerated by exploiting task level parallelism. As shown in fig. 2, multiple FE tasks can be executed in parallel, and the FD and DG subtasks within each FE task can also be executed in parallel.

However, the downside is that each set of tiles must be completed before the next set of tiles can be selected. This is enforced by a synchronization barrier, which blocks execution until all feature extraction tasks have completed.

B. Feature Extraction Task Execution Time

The execution time of the feature extraction task depends on two factors. First, the workload of the task itself can vary depending on the number of features that are detected from each tile, N_F. Second, the communication time can vary depending on the amount of congestion on the communication medium. Thus the FE task execution time, which consists of the computation time and the communication time, can be expressed as

$$t_{FE} = t_{comp} + t_{comm}, \quad (1)$$

$$\text{where } t_{comp} = (\alpha + N_F\beta), \quad (2)$$

$$\text{and } t_{comm} = \gamma(\delta + N_F\varepsilon). \quad (3)$$

Here, α, β, δ, and ε are constants that are algorithm dependent. α and β are the offset and proportional computation times, respectively, while δ and ε are the offset and proportional communication time, respectively. More interesting, are the variables N_F and γ. N_F, as mentioned above, is the number of features in each tile. γ is the congestion factor whose value is equal to or greater than 1. γ = 1 for the best case with no congestion.

Since N_F depends on the image content and is out of our control, assuming we can approximately predict N_F, we may be able to equalize the execution times for concurrent FE tasks by controlling the congestion factor γ.

III. System Architecture

Figure 3 shows the overall hardware mapping of the UVAM. The main processing elements are organized into the Cognitive Control Layer (CCL) 4 Feature Extraction Clusters (FEC), and a Feature Matching Processor (FMP). Two bidirectional off-chip gateways, not explicitly shown in fig.3, provide access to off-chip memories and a host controller.

The CCL consists of a RISC processor, the Intelligent Inference Engine (IIE) [10], the Visual Attention Engine (VAE), the power mode controller (PMC), and some fixed function units for accelerating visual attention. The IIE is a versatile mixed mode adaptive neuro-fuzzy inference system (ANFIS) [11] that was previously demonstrated to be capable of familiarity inference and power mode prediction [8]. In this work, the IIE is also used for FE task workload prediction as will be explained in section IV.

The 4 Feature Extraction Clusters (FEC) are capable of executing 4 FE tasks in parallel. Each FEC consists of 1 Vector Processing Element (VPE) for the FD task and 8 Scalar Processing Elements (SPE) for the DG task. The VPE is a 20-way SIMD processor optimized for the data-parallel operations of the FD task. The SPE is a 16b scalar processor with special instructions optimized for the DG task.

A total of 51 IPs are connected by a 2-level hierarchical star augmented by a ring network connecting the FECs' local routers, as shown in fig. 4. The hierarchical star network provides low latency, high bandwidth within each local network, as well as a low maximum hop count of 3 between

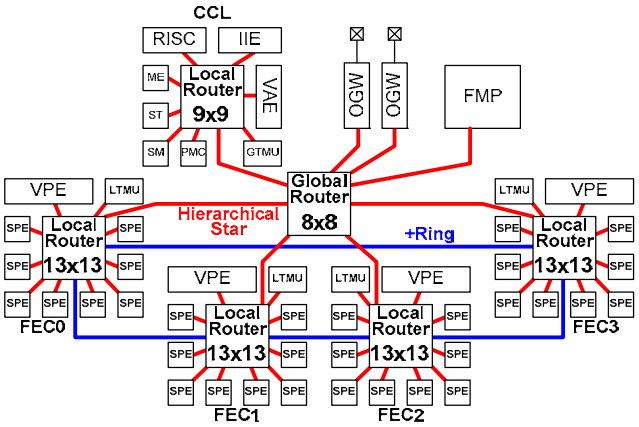


Figure 4. Hierarchical Star + Ring NoC topology.

any two nodes in the network. The ring network provides additional bandwidth for inter-FEC communication.

IV. Intelligent Workload Balancing

A. Neuro-Fuzzy Workload Prediction

It was shown in equations (1)~(3) that if the number of features N_F of each FE task can be predicted, then the execution time of concurrent FE tasks can be balanced by controlling the congestion factor γ. Normally, N_F is only known after executing the FD subtask, which consists of complex operations such as Gaussian pyramid generation and local maximum search. However, since N_F is closely dependent on the texture content, the IIE can be used to predict N_F using three simple statistics as shown in fig. 5. The variance of pixel intensities, and the peak counts from the vertical and horizontal projections of a tile provide clues on the amount of texture contained in the scene. The IIE uses fuzzy-logic rules that have been previously learned to infer N_F. It should be noted that the IIE need not predict the exact value of N_F. In most cases it is enough to correctly predict the relative values of N_F, since the priority weights assigned to each FEC are relative values as well.

B. Weighted Round Robin Bandwidth Regulation

Most network congestion occurs for traffic converging to the off-chip memory. Therefore the congestion factor γ can be controlled by regulating the bandwidth of traffic going through the off-chip gateways (OGW). The global router

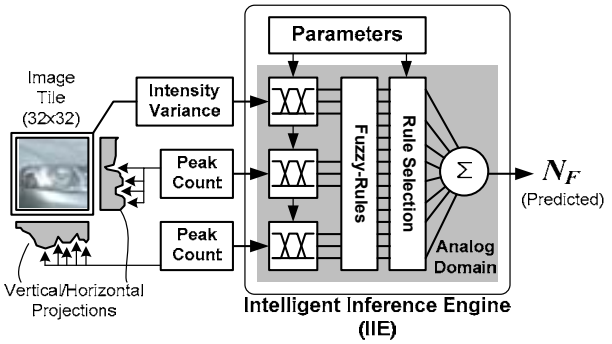


Figure 5. Predicting N_F using the IIE.

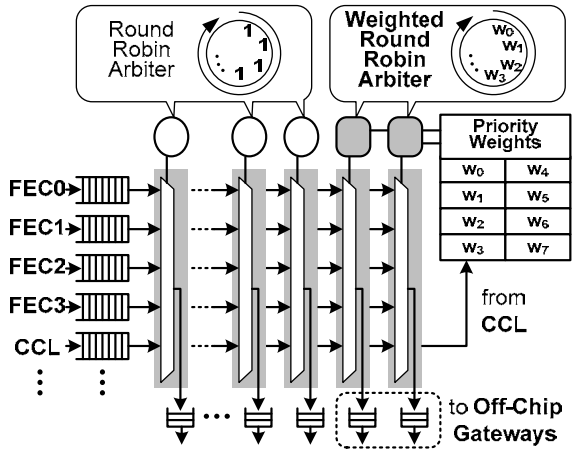


Figure 6. Global router architecture.

architecture is shown in fig. 6. The output ports to the OGWs are controlled by weighted round robin arbiters [9]. In weighted round robin arbitration, input ports with higher priority are granted a proportionately higher bandwidth. Therefore, for FE tasks with large N_F, a high priority weight is assigned for the corresponding FEC channel. This effectively reduces the congestion factor γ to balance out the execution time among FE tasks. The register file holding the priority weights are accessed directly by the CCL input port.

Simulation results of the proposed bandwidth regulation are shown in fig. 7. The data transfer rates between 4 FECs and the off-chip memory are plotted against time with and without bandwidth regulation. It can be seen that the transfer rate of the off-chip memory(synchronous SRAM) maxes out near 800 MB/s. When all of the FECs contend for off-chip memory bandwidth without bandwidth regulation, it can be seen that each FEC gets about 25% or 200 MB/s of the available bandwidth. However, with bandwidth regulation enabled, FEC3, which would have otherwise finished last, is assigned the highest priority weight and finishes first. As a result, in this example the overall execution time of the 4 FECs is reduced by 12.1%.

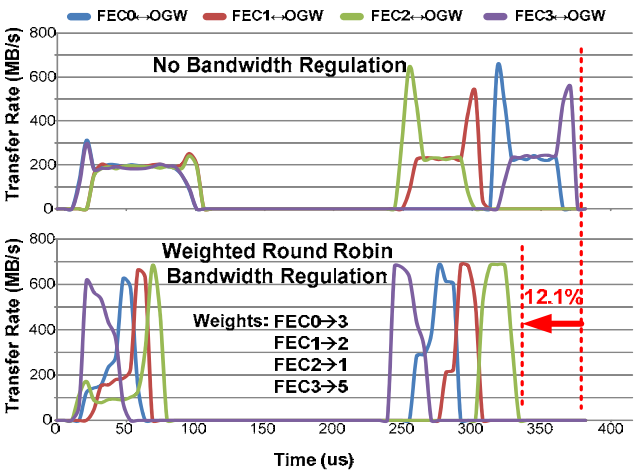
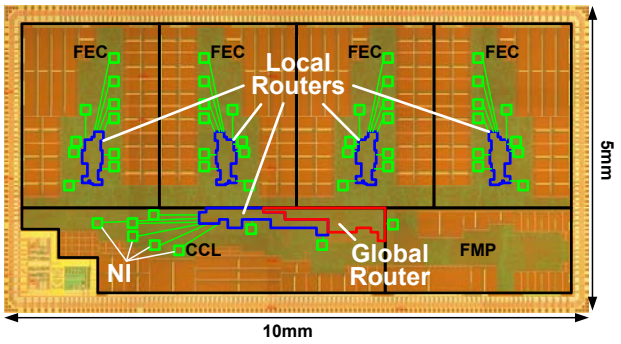


Figure 7. Bandwidth regulation results.



Process	0.13µm 8LM Logic CMOS
Die Size	10mm x 5mm
Gates/SRAM	2.93M Gates, 612kB
# of Nodes	51
Supply Voltage	Global, CCL: 1.2V / FEC: 0.65~1.2V
Frequency	Global:400MHz / CCL:200MHz / FEC: 50~200MHz
Aggregate BW	76.8GB/s
Power	42mW(NoC), 345mW(Chip)
NoC Features	NI w/ 2D DMA, Wormhole Routing,

Fig. 8. Chip photograph and NoC summary.

V. Implementation Results

The heterogeneous multi-core object recognition chip[8], shown in fig. 8, is implemented in a 0.13um CMOS process and contains 2.93M gates and 612kB of SRAM. The NoC and all IPs, excluding the analog portion of the IIE and level shifting circuits for power domain crossing, are implemented in a standard cell PnR flow. The location of the PnR'ed routers and network interfaces (NI) are shown in fig. 8.

The object recognition processor has been integrated within a augmented reality headset as shown in fig. 9. 30fps VGA video is received from the VGA camera and processed by the NoC based object recognition processor in real time. Information about the recognized objects are overlaid on the output video displayed on the head mounted display (HMD). Thanks to the UVAM and workload balancing of the intelligent NoC, a low power consumption of 345mW is achieved. The neuro-fuzzy prediction based bandwidth regulation was responsible for an average 34% reduction in the idle time and 11.5% reduction in execution time of the FECs. The measured output voltage of the IIE performing N_F prediction is shown in fig. 10. The evaluation time is only 250ns and negligible compared to the FE task.

VI. Conclusion

An intelligent NoC that performs workload prediction and workload balancing to accelerate object recognition in a 51 IP SoC has been realized. Accurate content-based workload prediction is performed by the IIE, which exploits the correlation between texture content and the number of features, N_F, in each 32x32 pixel tile. The execution time of 4 FECs, which each process 1 tile, is balanced by a weighted round robin packet arbitration scheme that gives priority to FECs that are assigned tiles with higher predicted N_F. As a result, 11.5% average reduction in execution time was achieved. This proves that NoC channel arbitration controlled by neuro-fuzzy inference is an effective method for workload balancing in a multi-core processor.

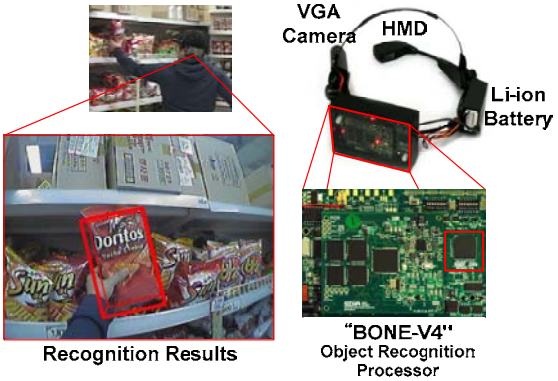


Fig. 9. Augmented reality headset demonstration.

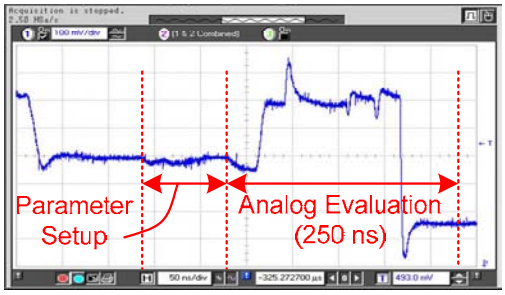


Fig. 10. Measured waveform of IIE inference output.

References

[1] Kyo, S., and Okazaki, S., "IMAPCAR: A 100 GOPS In-Vehicle Vision Processor Based on 128 Ring Connected Four-Way VLIW Processing Elements," Journal of Signal Processing Systems, DOI 10.1007/s11265-008-0297-0

[2] Kim, D., et al., 2007, "An 81.6 GOPS Object Recognition Processor Based on NoC and Visual Image Processing Memory," Proc. of IEEE CICC 2007, pp. 443-446.

[3] Kim, K., et al., 2008, "A 125GOPS 583mW Network-on-Chip Based Parallel Processor with Bio-inspired Visual-Attention Engine," 2008 ISSCC Dig. Tech. Papers, pp. 308-309.

[4] Kim, J.-Y., et al., 2009, "A 201.4GOPS 496mW Real-Time Multi-Object Recognition Processor with Bio-Inspired Neural Perception Engine," 2009 ISSCC Dig. Tech. Papers, pp. 150-151.

[5] Itti, L., Koch, C., and Niebur, E., 1998, "A model of saliency-based visual attention for rapid scene analysis," IEEE Trans. PAMI, vol.20, no. 11, pp. 124-1259.

[6] Lee, S., et al., 2010, "Familiarity based unified visual attention model for fast and robust object recognition," Pattern Recognition, vol.43, pp.1116-1128.

[7] Lubachevsky, B., "Synchronization barrier and related tools for shared memory parallel programming," Intl. J. Par. Prog., vol. 19, no. 3, pp. 225-250.

[8] Lee, S., et al., 2010, "A 345mW Heterogeneous Many-Core Processor with an Intelligent Inference Engine for Robust Object Recognition," 2010 ISSCC Dig. Tech. Papers, pp.332-333.

[9] H.-J. Yoo, et al., "Low-Power NoC for High-Performance SoC Design," CRC Press, 2008.

[10] Oh, J., et al., 2010, "A 1.2mW On-Line Learning Mixed Mode Intelligent Inference Engine for Robust Object Recognition", 2010 Symp. VLSI Circuits 2010, accepted for publication.

[11] J.-S.R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 65-685, May 1993.